

Window Programming

Visual C++ MFC Programming

Lecture 01

김예진

Dept. of Game Software

Notices

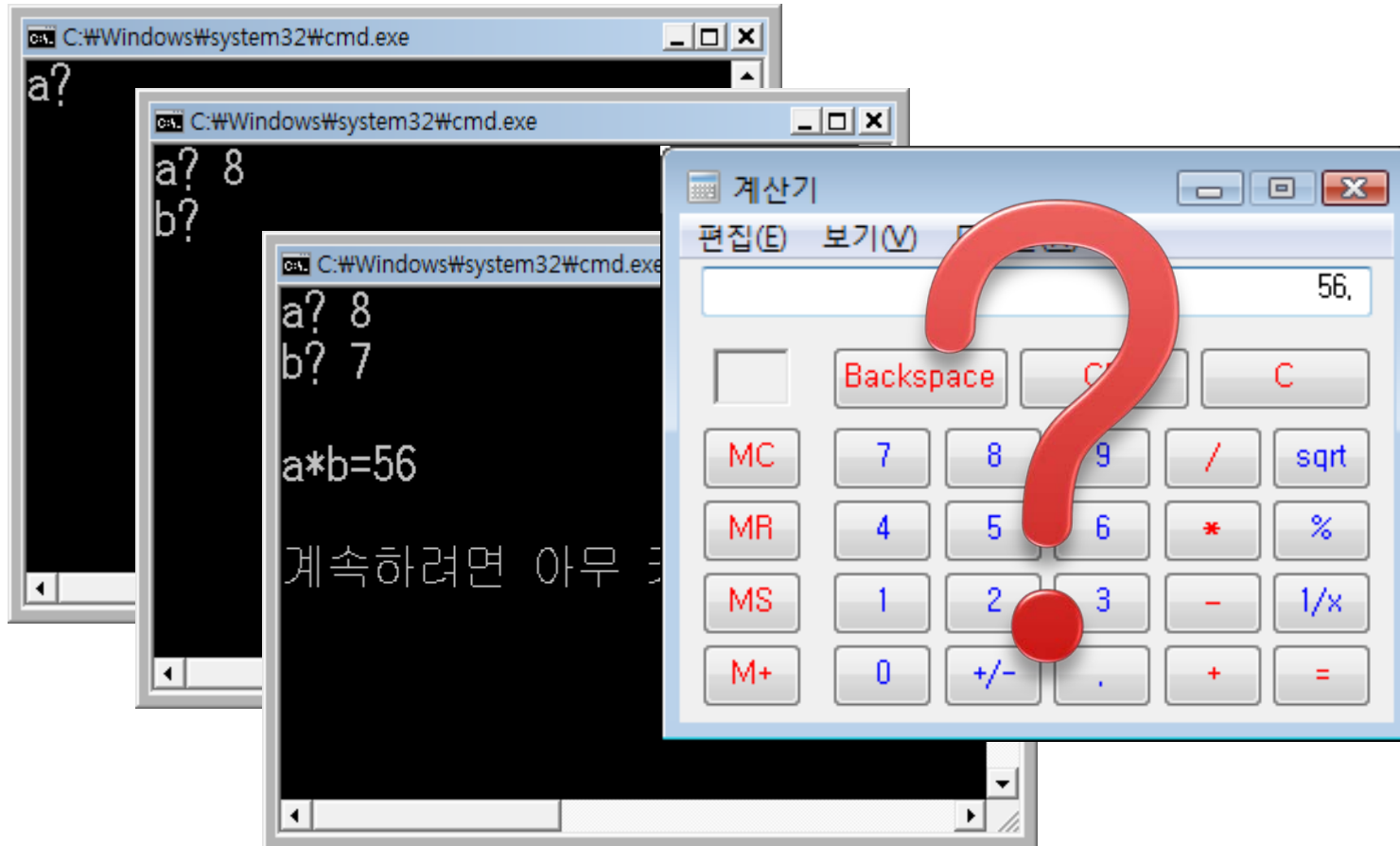
- 03/07: 502 → 501 등록 이동

Plan

- Window Programming 소개
- Visual C++ IDE (통합 개발 환경) 소개
 - MFC 응용 프로그램 생성

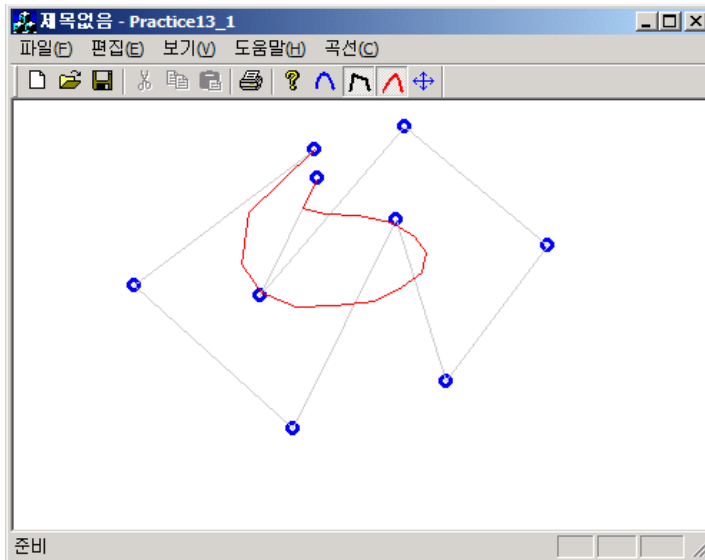
Window Programming 소개

- 지금까지 해왔던 코딩
 - C/C++ console programming



Window Programming 소개

- Window 프로그램 vs. Console (DOS) 프로그램
 - Code의 실행 순서에 따른 차이



```
C:\WINDOWS\system32\cmd.exe
C:\cygwin\home\hoon\work\netsim>dir/w
Volume in drive C is IBM_PRELOAD
Volume Serial Number is 40BA-B69E

Directory of C:\cygwin\home\hoon\work\netsim

[.]           [..]           a.exe         [data]        [include]     [lib]
Makefile      Netsim.exe  netsim.IAB   netsim.IAD   netsim.IMB   netsim.IMD
netsim.PFI   netsim.PO   netsim.PR   netsim.PRI   netsim.PS    netsim.WK3
[src]

               13 File(s)        109,256 bytes
                  6 Dir(s)      11,702,415,360 bytes free

C:\cygwin\home\hoon\work\netsim>netsim
Start netsim
Total coordinate X = 1500, Y = 300
Success in init the map.
node[0] => ( x = 433, y = 96, CID = 1 )
node[1] => ( x = 641, y = 197, CID = 7 )
node[2] => ( x = 209, y = 103, CID = 5 )
node[3] => ( x = 969, y = 53, CID = 3 )
node[4] => ( x = 821, y = 202, CID = 12 )
node[5] => ( x = 290, y = 81, CID = 0 )
node[6] => ( x = 393, y = 182, CID = 6 )
node[7] => ( x = 1069, y = 274, CID = 13 )
node[8] => ( x = 93, y = 180, CID = 5 )
node[9] => ( x = 1143, y = 17, CID = 3 )
node[10] => ( x = 1292, y = 80, CID = 4 )
node[11] => ( x = 603, y = 73, CID = 2 )
node[12] => ( x = 362, y = 236, CID = 11 )
```

Window Programming 소개

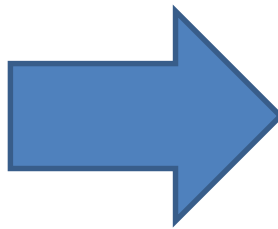
- 윈도우 프로그래밍이란?
 - 사용자가 발생시키는 **event**에 대한 **message**를 처리하는 것
 - 이벤트 기반 (Event-driven) or 메시지 기반 (Message-driven) 프로그래밍
- Event
 - 입력장치의 상태 변화 또는 내부의 상태 변화
- Message
 - 프로그램의 외부 또는 내부의 변화를 해당 프로그램에 알리는 개념

MOUSE EVENT HANDLERS starring HANDLERMAN



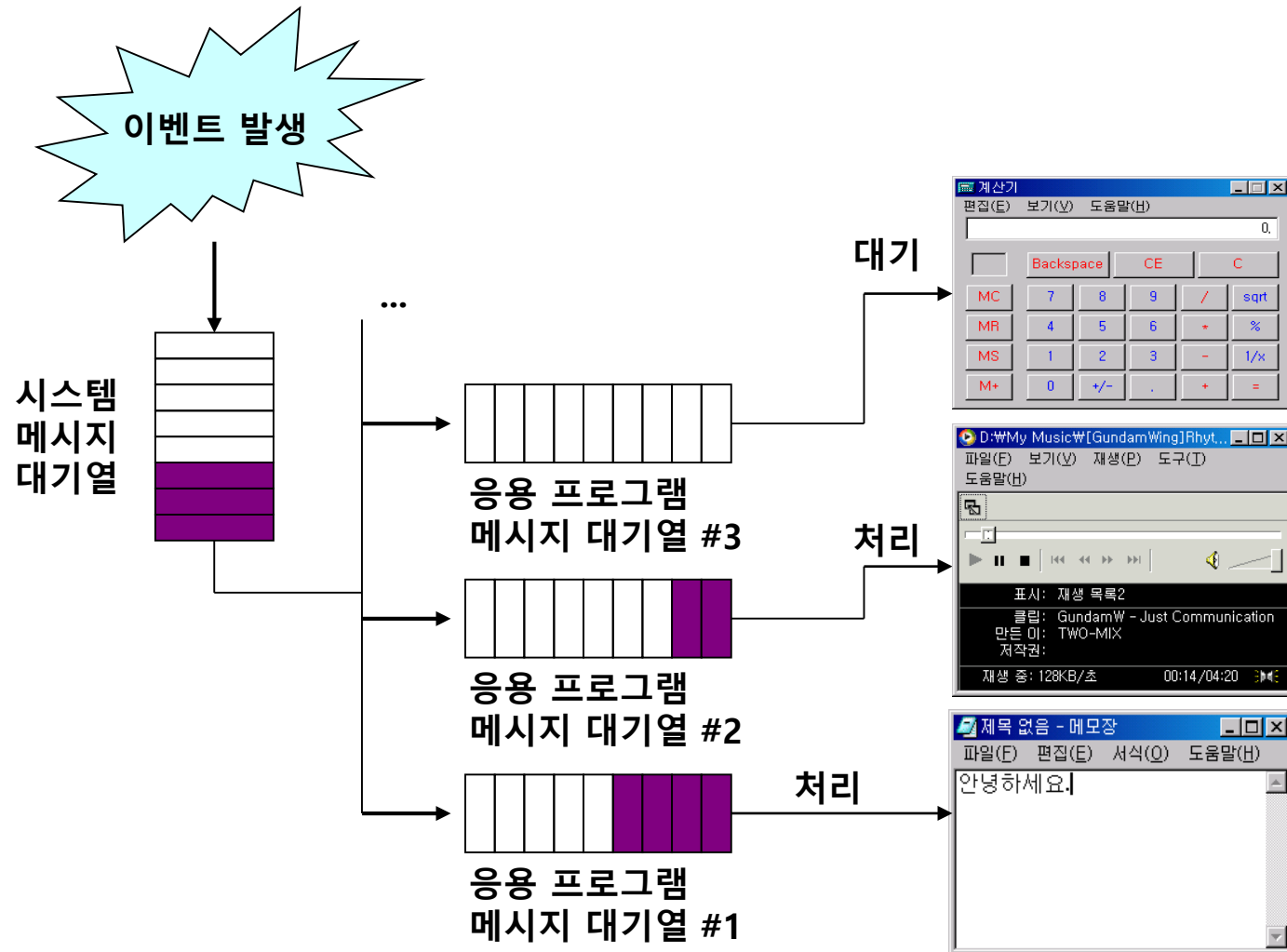
Window Programming 소개

- Event 기반 프로그래밍
 - Event가 발생되었을 때 이를 처리하는 방법을 알려 주는 것
 - Event handler: event를 처리하는 함수 (function)



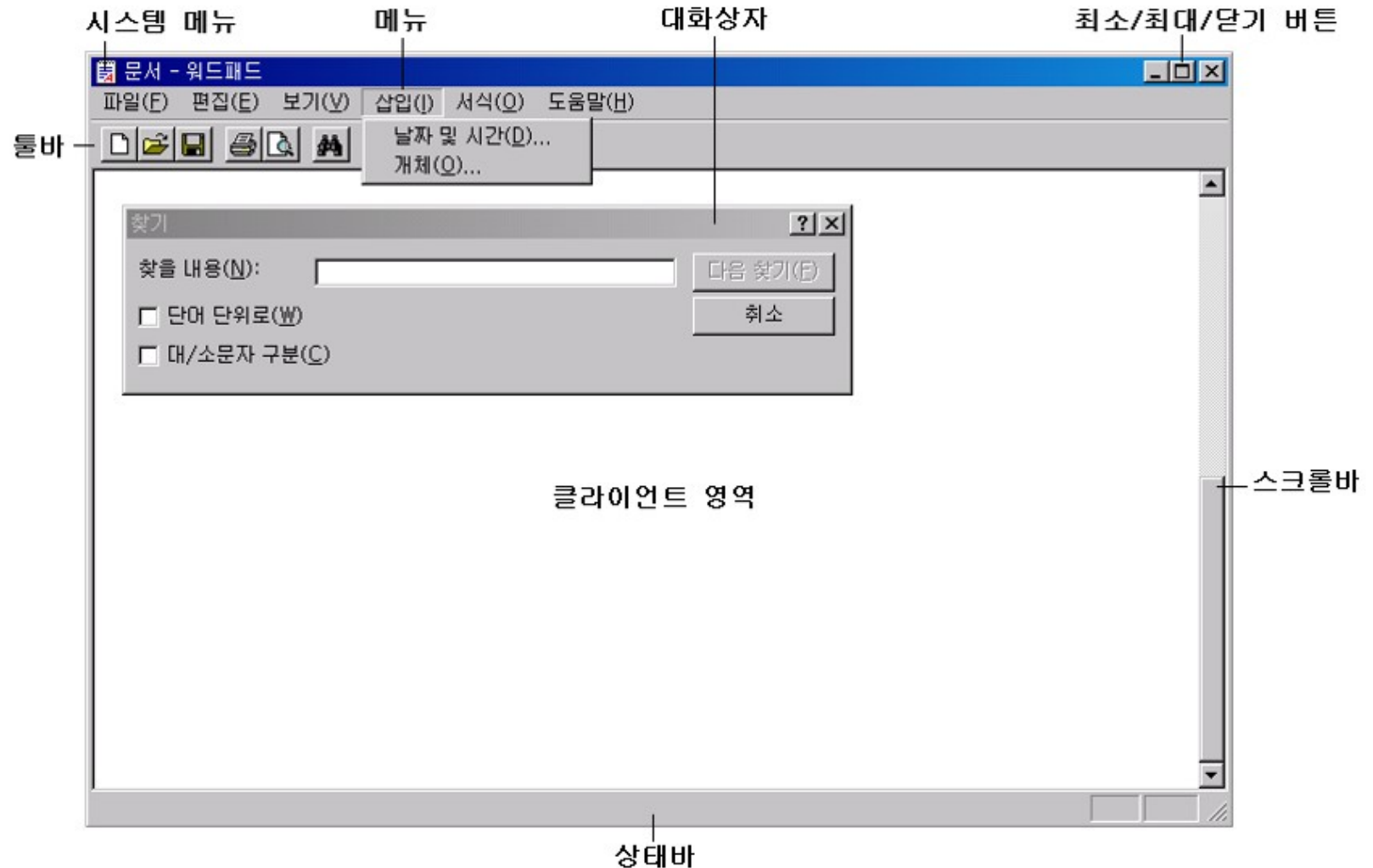
Window Programming 소개

- Event와 event handler



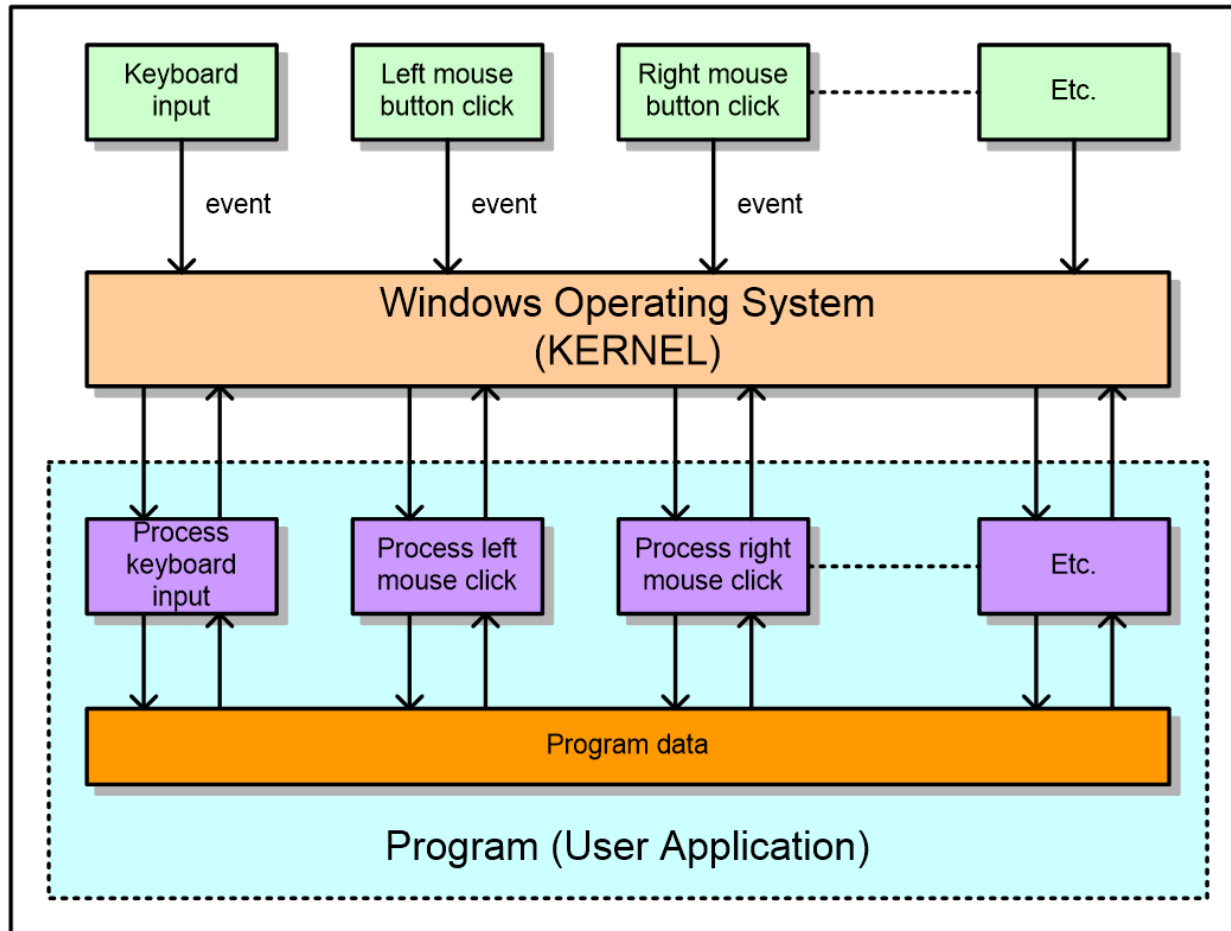
Window Programming 소개

- Graphical User Interface (GUI)



Window Programming 소개

- Window 프로그래밍의 기본 구조



Window Programming 소개

- API (Application Programming Interface) 기반
 - 운영체제가 응용 프로그램을 위해 제공하는 각종 함수의 집합
 - C 언어 기반
- 장점
 - 세부적인 제어가 가능
 - 윈도우 운영체제가 제공하는 모든 기능을 사용 가능
 - 생성 코드의 크기가 작고 속도가 빠름
- 단점
 - 생산성이 낮음

*SDK(Software Development Kit): 컴파일러, 개발툴, 소스코드, 헤더, 라이브러리, 도움말 등 윈도우 응용 프로그램 제작에 필요한 집합

Window Programming 소개

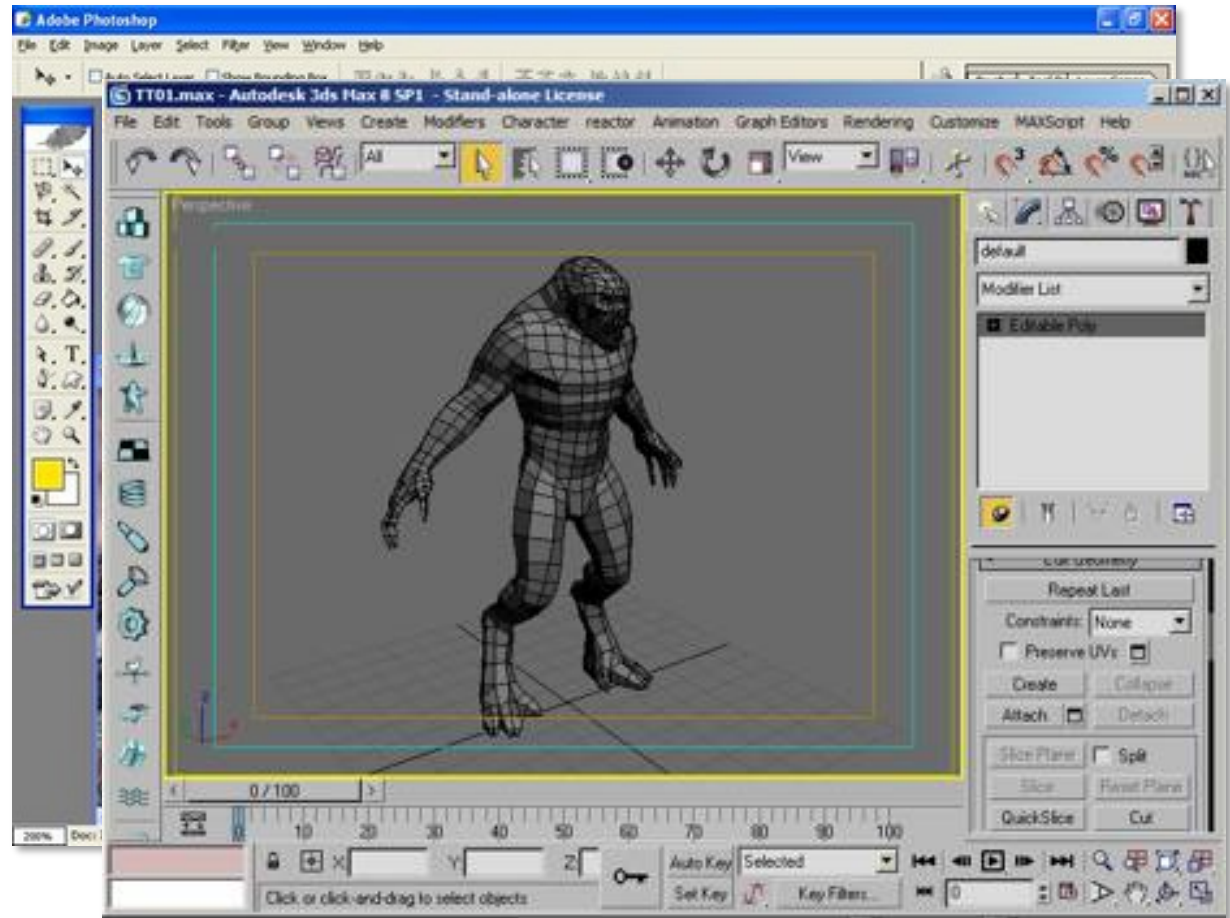
- RAD (Rapid Application Development) 기반
 - 시각적으로 화면을 디자인하고 코드를 추가하는 방법으로 프로그램을 빠르게 개발 가능
 - Visual Basic, Delphi 등
- 장점
 - 생산성이 높음
- 단점
 - 일반적으로 생성 코드의 크기가 크고 실행 속도가 느림
 - 운영체제가 제공하는 모든 기능을 활용한 세부적인 제어가 어려움

Window Programming 소개

- MFC (Microsoft Foundation Class Library) 기반
 - API 중 많이 사용되는 기능들을 사용하기 쉽게 잘 포장해 놓은 것
 - 객체 지향 언어를 이용하여 클래스로 제공
- 장점
 - API 직접 이용 보다 생산성이 높음
 - RAD보다 생성 코드의 크기가 작고 실행 속도가 빠름
- 단점
 - 초기 학습 시간이 긴 편
 - 객체 지향 언어
 - 클래스 라이브러리 구조와 기능 및 각 클래스의 관계 파악

Window Programming 소개

- C/C++ 프로그래밍
 - 변수/배열
 - =, +, -, *, /
 - if
 - for
 - function
 - Pointer
 - **class**
 - **template**



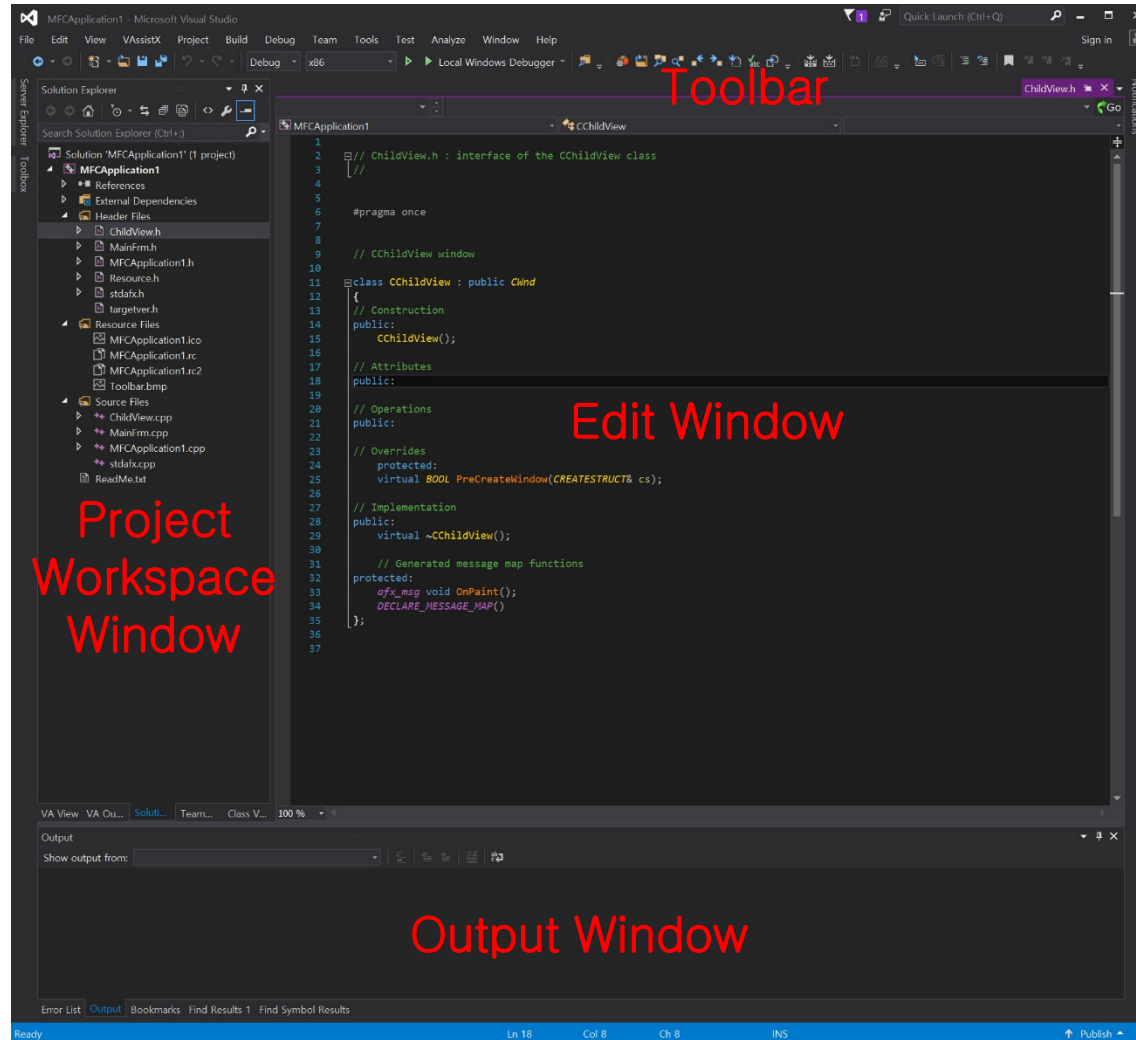
Window Programming 소개

- C++의 기초 필요
 - 객체지향 언어 (object-oriented language)
 - 클래스 (class), 상속 (inheritance), 다형성 (polymorphism), 가상 함수 (virtual functions)
- 자신만의 친숙한 개발 도구 선택
 - Microsoft Visual Studio (5.0~6.0, 2002~2013, 2015~2017)
 - 본 과목에서는 **Visual Studio 2017** 에서 제공하는 통합 개발 환경 (IDE) 사용



VC++ IDE

- Visual Studio interface

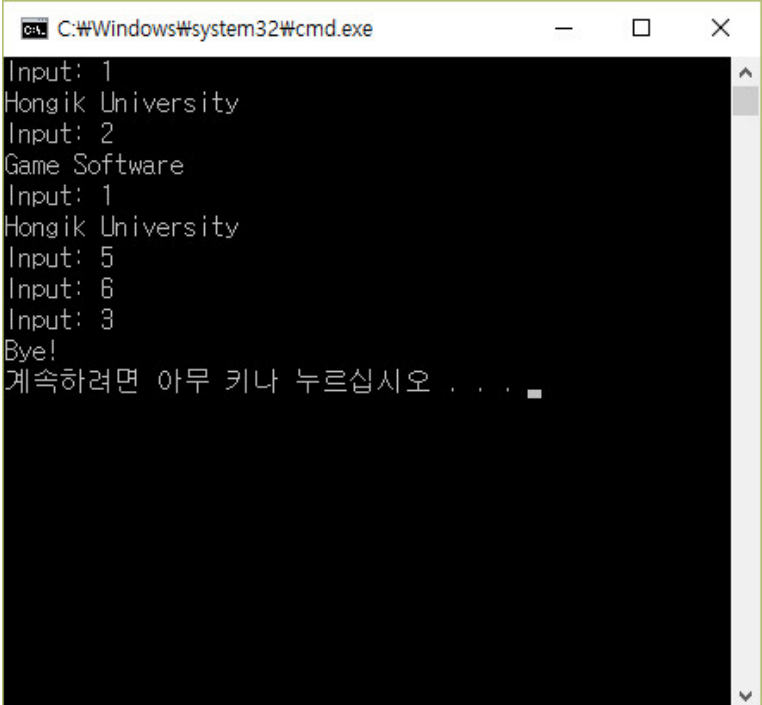


VC++ IDE

- 시스템의 구성 요소
 - Editor
 - Source code 작성
 - Compile, editing, debugging 등의 interactive 환경 제공
 - Compiler
 - Compile : Source code → Machine language
 - Error and warning detecting and notifying
 - Output : object code (.obj)
 - Linker
 - Link : module merging + program libraries
 - Linking error detecting and notifying
 - Library
 - Standard library
 - MFC & T (Microsoft Foundation Classes and Templates)
- 개발 과정 단순화를 위한 기타 도구
 - **AppWizard** : Windows 프로그램의 기본적인 Framework 자동 생성
 - **ClassWizard** : AppWizard에 의해 생성된 클래스 확장 용이성 제공

간단한 코딩 연습

- C++를 사용하여, 사용자의 입력에 따라
 - '1'을 입력하면 "Hongik University"를 출력
 - '2'를 입력하면 "Game Software"를 출력
 - '3'을 입력하면 "Bye~"를 출력하고 종료
 - 위의 과정을 무한 반복



```
C:\Windows\system32\cmd.exe
Input: 1
Hongik University
Input: 2
Game Software
Input: 1
Hongik University
Input: 5
Input: 6
Input: 3
Bye!
계속하려면 아무 키나 누르십시오 . . .
```

간단한 코딩 연습

- C++ Hints

- 출력: printf 대신 std::cout 사용

```
std::cout << "Software" << std::endl;
```

- 입력: scanf 대신 std::cin 사용

```
int i;  
std::cin >> i;
```

- 무한 반복

```
while (true)  
{  
...  
}
```

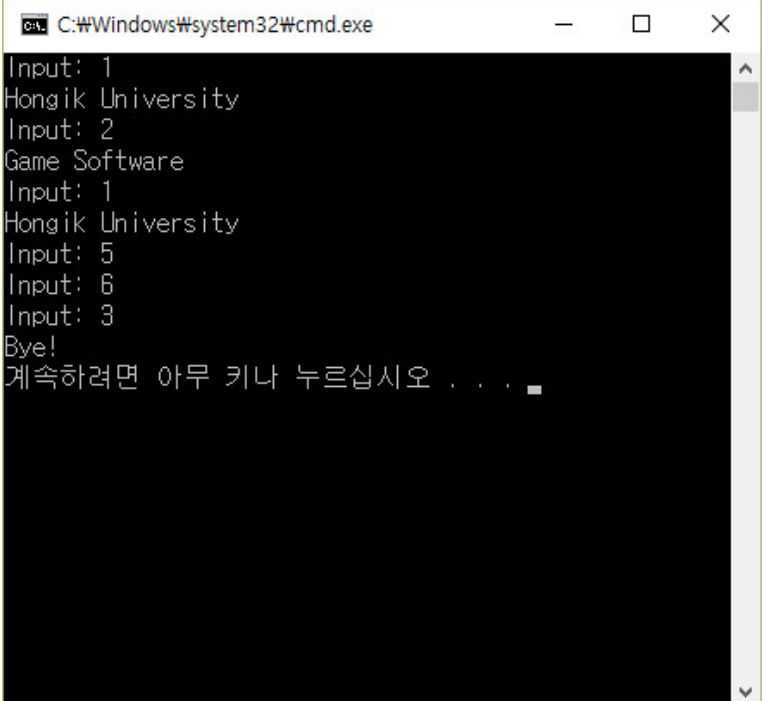
- 선택

```
switch (i)  
{  
    case 1:  
        ...  
        break;  
  
    default:  
        ...  
        Break;  
}
```

- *iostream header file 추가 필요

```
// std::cin, std::cout, std::endl 사용시  
#include <iostream>
```

.h 확장자가 없는 file은 1998년 제정된 C++ 표준을 준수하는 file임



```
C:\Windows\system32\cmd.exe  
Input: 1  
Hongik University  
Input: 2  
Game Software  
Input: 1  
Hongik University  
Input: 5  
Input: 6  
Input: 3  
Bye!  
계속하려면 아무 키나 누르십시오 . . .
```

코딩 예

```
#include <iostream>

int main()
{
    int i;
    while (true){
        std::cout << "Input: ";
        std::cin >> i;

        switch (i){
            case 1:
                std::cout << "Hongik University" << std::endl;
                break;
            case 2:
                std::cout << "Game Software" << std::endl;
                break;
            case 3:
                std::cout << "Bye!" << std::endl;
                return 0;
            default:
                break;
        }
    }
    return 0;
}
```

코딩 예

```
#include <iostream>
using namespace std;

int main()
{
    int i;
    while (true){
        cout << "Input: ";
        cin >> i;

        switch (i){
            case 1:
                cout << "Hongik University" << endl;
                break;
            case 2:
                cout << "Game Software" << endl;
                break;
            case 3:
                cout << "Bye!" << endl;
                return 0;
            default:
                break;
        }
    }
    return 0;
}
```

간단한 코딩 연습

Message (Event)

사용하여, Message (Event) Handler 따라

Windows
Procedure

- '1'을 입력하면 "Hongik University"를 출력
- '2'를 입력하면 "Game Software"를 출력
- '3'을 입력하면 "Bye~"를 출력하고 종료

- 위의 과정을 무한 반복

Message Loop

```
C:\Windows\system32\cmd.exe
Input: 1
Hongik University
Input: 2
Game Software
Input: 1
Hongik University
Input: 5
Input: 6
Input: 3
Bye!
계속하려면 아무 키나 누르십시오 . . .
```

코딩 예

```
#include <iostream>
using namespace std;

int main()
{
    int i;
    while (true){
        cout << "Input: ";
        cin >> i;
        switch (i){
            case 1:
                cout << "Hongik University" << endl;
                break;
            case 2:
                cout << "Game Software" << endl;
                break;
            case 3:
                cout << "Bye!" << endl;
                return 0;
            default:
                break;
        }
    }
    return 0;
}
```

Message
Loop

Message
Handler

Message

좀 더 멋있게...

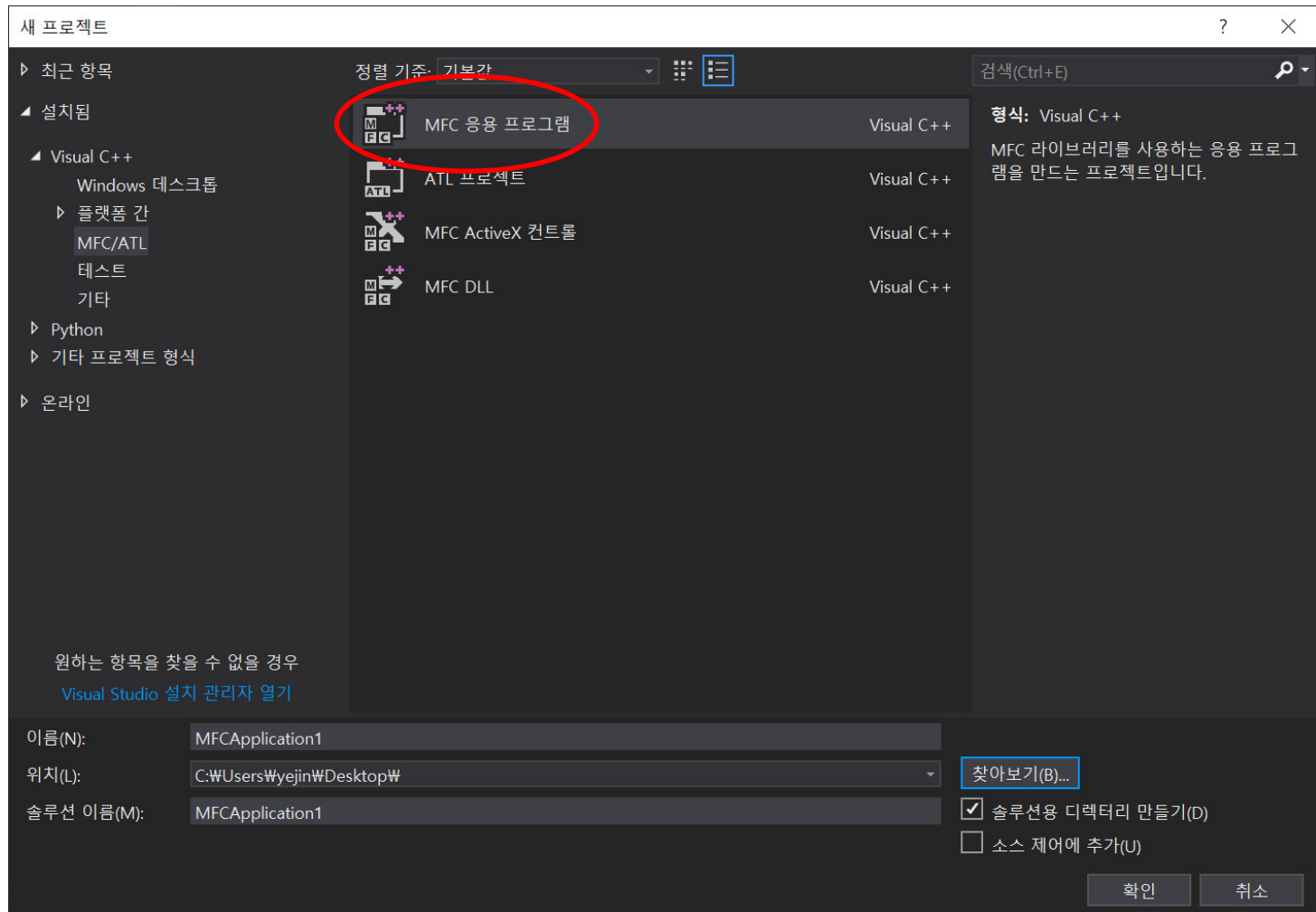
```
int main()
{
    int i;

    while(true)
    {
        cout << "Input: ";
        cin >> i;
        procedure(i);
    }
    return 0;
}
```

```
void procedure(int msg)
{
    switch(msg)
    {
        case 1:
            cout << "Hongik University" << endl;
            break;
        case 2:
            cout << "Game Software" << endl;
            break;
        case 3:
            cout << "Bye!" << endl;
            exit(0);
            break;
        default:
            break;
    }
}
```


MFC 응용 프로그램 생성

- 프로젝트 종류 선택: AppWizard



MFC 응용 프로그램 생성

- AppWizard 기능
 - 만들고자 하는 기본적인 project를 생성
 - Project 안에 필요한 class 생성
 - class에 기본적인 내용을 자동 코딩
 - 기본적인 코딩시간을 절약 하므로 빠른 project 완성
 - AppWizard사용 도중 실수로 option을 선택하지 않았을 경우 source code에 직접 새로 추가할 수 있음

MFC 응용 프로그램 생성

- AppWizard 1단계

MFC 응용 프로그램
응용 프로그램 종류 옵션

응용 프로그램 종류

응용 프로그램 종류(T):
단일 문서

응용 프로그램 종류 옵션:

☐ 템플릿 문서(B)

☐ 문서 뷰 아키텍처 지원(V)

☒ SDI(Security Development Lifecycle) 검사(C)

대화 상자 기반 옵션(I):
<없음>

복합 문서 지원:
<없음>

문서 지원 옵션:

☐ 활성 문서 서버(A)

☐ 활성 문서 컨테이너(D)

☐ 복합 파일 지원(U)

프로젝트 스타일:
MFC standard

비주얼 스타일 및 색(Y):
Windows Native/Default

☐ 비주얼 스타일 전환 사용(C)

리소스 언어(L):
English (United States)

MFC 사용:
공유 DLL에서 MFC 사용

이전 다음 마침 취소

MFC 응용 프로그램 생성

- AppWizard 2단계

MFC 응용 프로그램

문서 템플릿 속성

응용 프로그램 종류

문서 템플릿 속성

사용자 인터페이스 기능

고급 기능

생성된 클래스

파일 확장명(X):

파일 형식 처리기 옵션:

☐ 미리 보기 처리기(V)

☐ 축소판 그림 처리기(B)

☐ 검색 처리기(H)

주 프레임 캡션(P):

MFCApplication1

문서 종류 이름(T):

MFCApplication1

파일의 새 약식 이름(S):

MFCApplication1

필터 이름(N):

파일 형식 ID(I):

MFCApplication1.Document

파일 형식의 긴 이름(L):

MFCApplication1.Document

이전

다음

마침

취소

MFC 응용 프로그램 생성

- AppWizard 3단계

MFC 응용 프로그램
사용자 인터페이스 기능

응용 프로그램 종류

문서 템플릿 속성

사용자 인터페이스 기능

고급 기능

생성된 클래스

주 프레임 스타일:

- ☒ 두꺼운 프레임(T)
- ☒ 최소화 상자(I)
- ☒ 최대화 상자(A)
- ☐ 최소화(N)
- ☐ 최대화(I)
- ☒ 시스템 메뉴(Y)
- ☐ 정보 상자(B)
- ☒ 초기 상태 표시줄(S)
- ☐ 분할 창(P)

자식 프레임 스타일:

- ☐ 자식 최소화 상자(M)
- ☒ Child maximize box
- ☐ 자식 창 최대화(D)

Command bar (menu/toolbar/ribbon):

클래식 메뉴 사용

Classic menu options:

<없음>

Menu bar and toolbar options:

- ☐ 사용자 정의 도구 모음 및 이미지(E)
- ☒ 개인 설정 메뉴 동작(H)

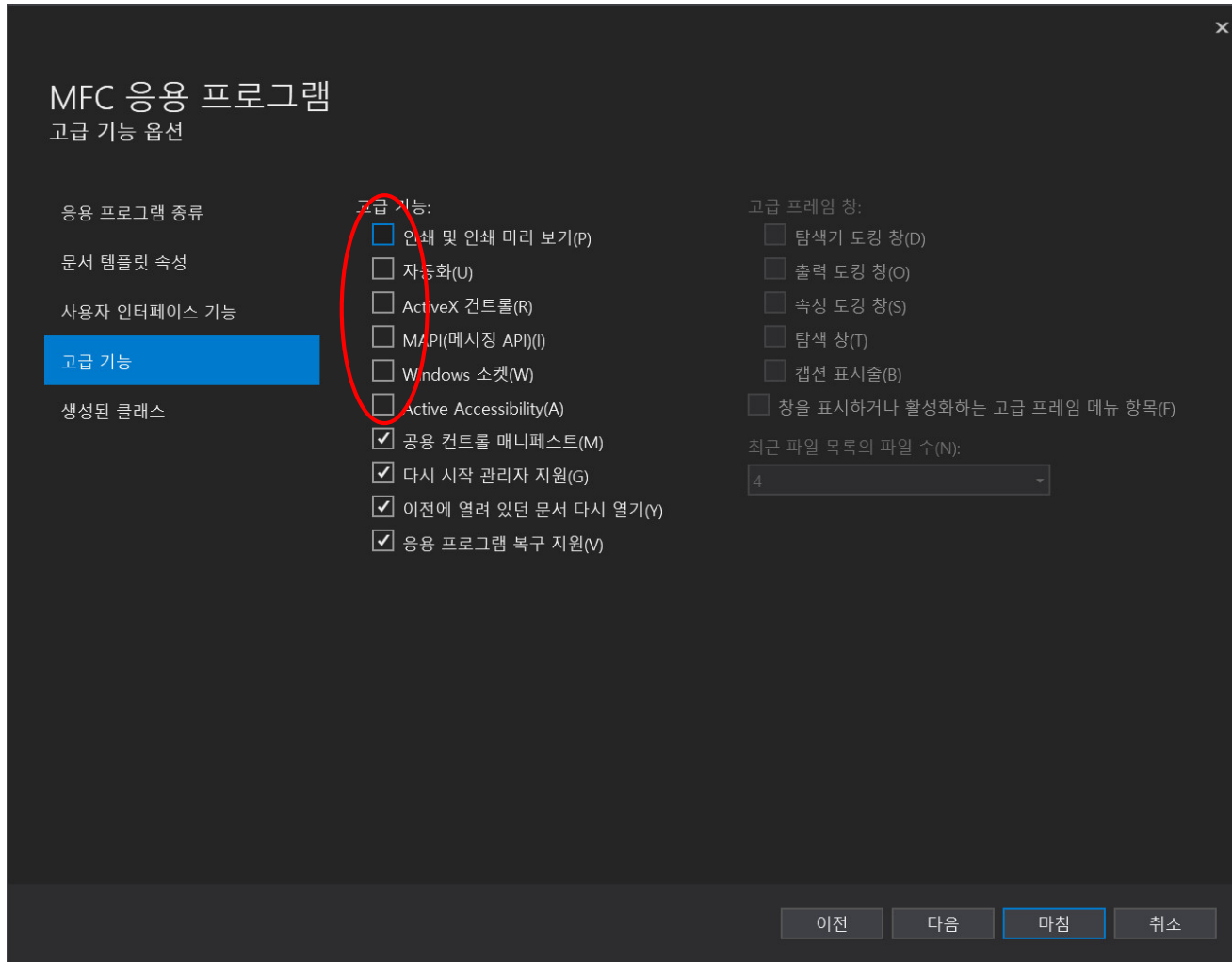
대화 상자 제목(G):

MFCApplication1

이전 다음 마침 취소

MFC 응용 프로그램 생성

- AppWizard 4단계



MFC 응용 프로그램 생성

- AppWizard 5단계

The screenshot shows the 'MFC 응용 프로그램 생성된 클래스 옵션' (MFC Application Generated Class Options) dialog box. It is the fifth step of the wizard. The left sidebar contains five options: '응용 프로그램 종류' (Application Type), '문서 템플릿 속성' (Document Template Properties), '사용자 인터페이스 기능' (User Interface Features), '고급 기능' (Advanced Features), and '생성된 클래스' (Generated Class), which is currently selected and highlighted in blue. The main area is divided into two columns. The left column contains three settings: '생성된 클래스(G):' (Generated Class) with a dropdown menu showing 'App', '클래스 이름(L):' (Class Name) with a text box containing 'CMFCApplication1App', and '기본 클래스(A):' (Base Class) with a dropdown menu showing 'CWinApp'. The right column contains two settings: '헤더 파일(H):' (Header File) with a text box containing 'MFCApplication1.h' and '.cpp 파일(P):' (CPP File) with a text box containing 'MFCApplication1.cpp'. At the bottom right, there are four buttons: '이전' (Previous), '다음' (Next), '마침' (Finish), and '취소' (Cancel). The '마침' button is highlighted in blue.

MFC 응용 프로그램
생성된 클래스 옵션

응용 프로그램 종류

문서 템플릿 속성

사용자 인터페이스 기능

고급 기능

생성된 클래스

생성된 클래스(G):
App

클래스 이름(L):
CMFCApplication1App


기본 클래스(A):
CWinApp

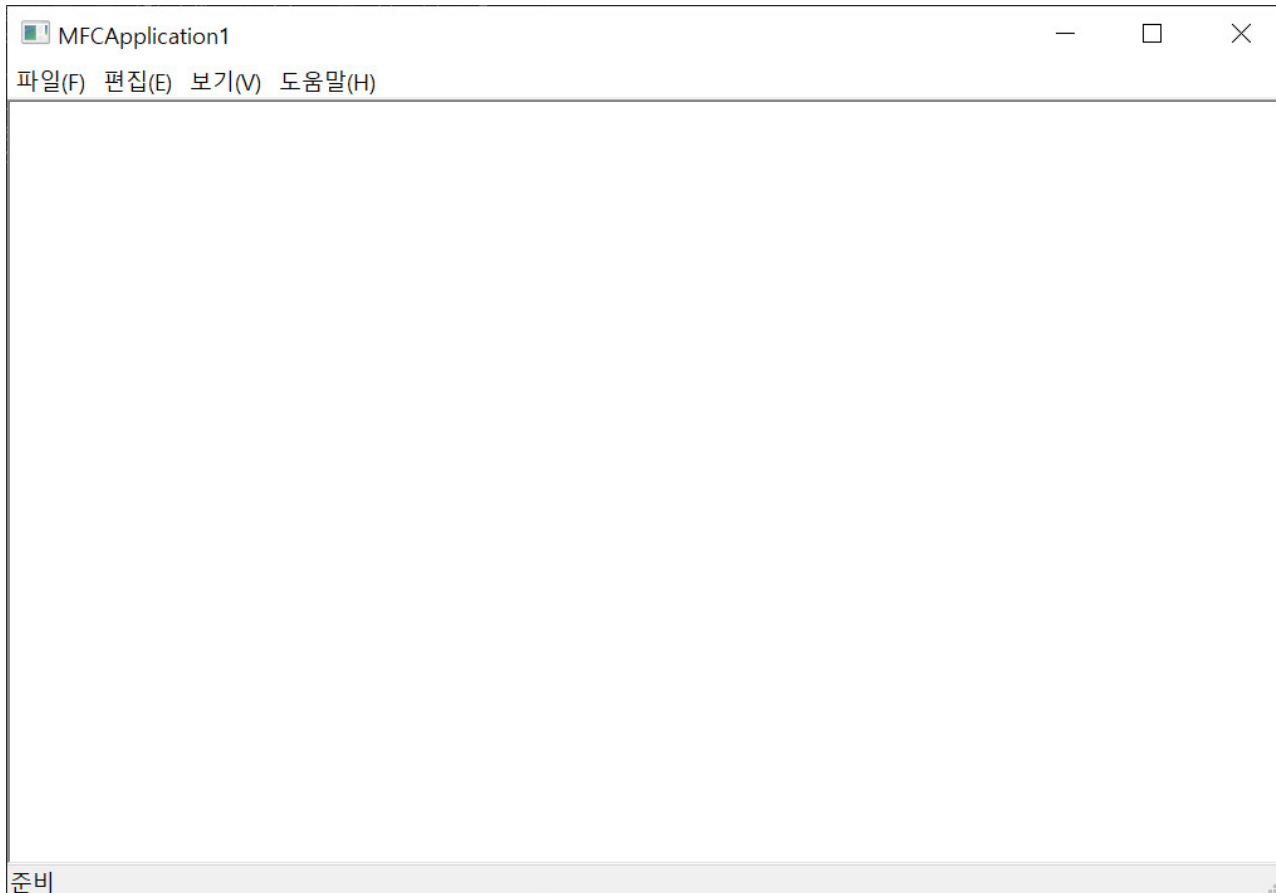
헤더 파일(H):
MFCApplication1.h

.cpp 파일(P):
MFCApplication1.cpp

이전 다음 마침 취소

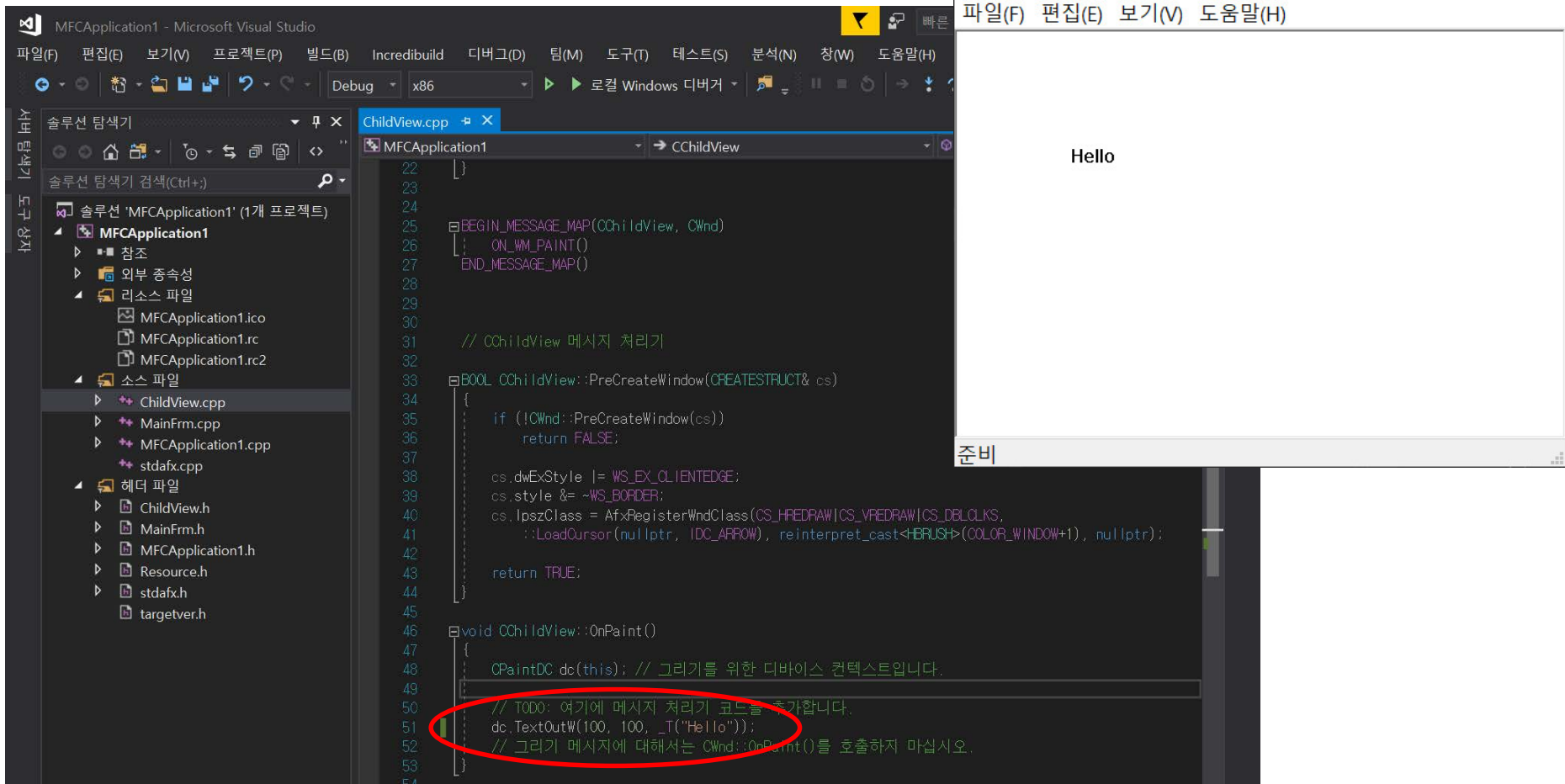
MFC 응용 프로그램 생성

- 실행
 - Ctrl + F5 or  로컬 Windows 디버거 ▾



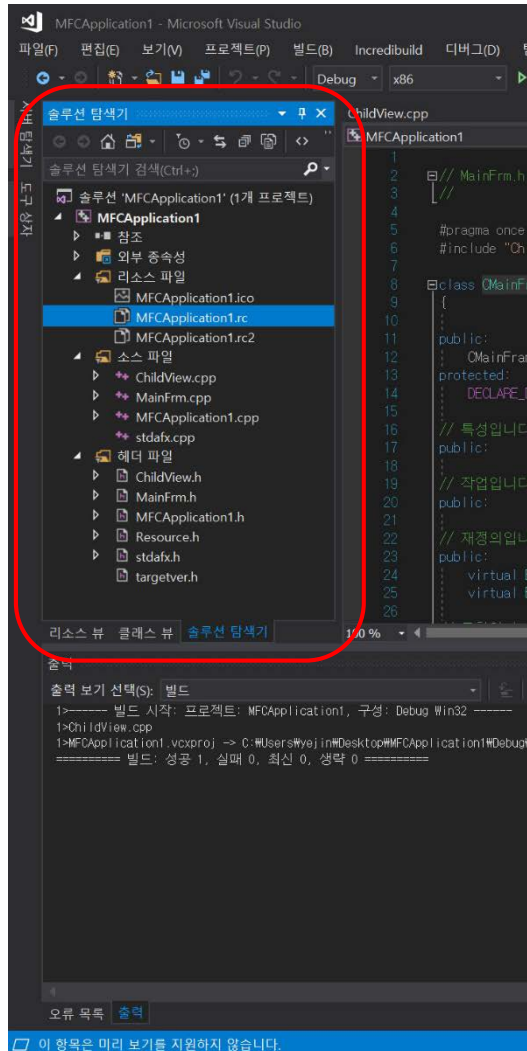
MFC 응용 프로그램 생성

- “void CChildView::OnPaint()”에 아래 코드 추가
dc.TextOut(100, 100, _T("Hello"));



MFC 응용 프로그램 생성

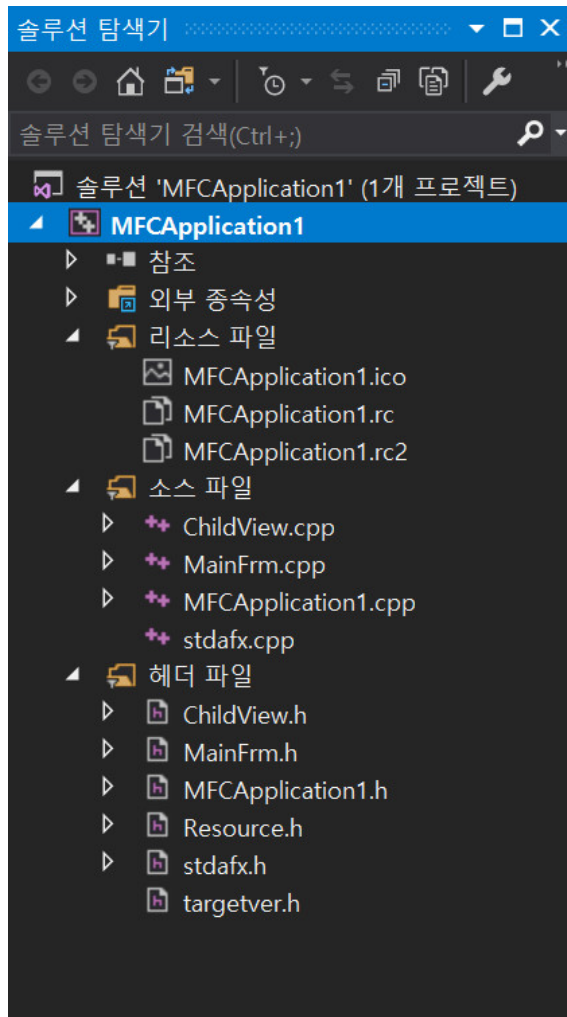
- Project Workspace의 항목별 설명



항목	내용
클래스 뷰 (Class View)	Project에 설정되어 있는 class별로 출력, 해당 항목을 선택하면 수정 가능
리소스 뷰 (Resource View)	Project에 설정되어 있는 메뉴, 대화 상자, 문자열, 아이콘, 비트맵 등 자원의 리스트 출력, 해당 항목 선택 수정 가능
솔루션 탐색기 (Solution Explore) (FileView)	Project에 설정되어 있는 파일 리스트 출력, 해당 항목을 선택하여 수정 가능

MFC 응용 프로그램 생성

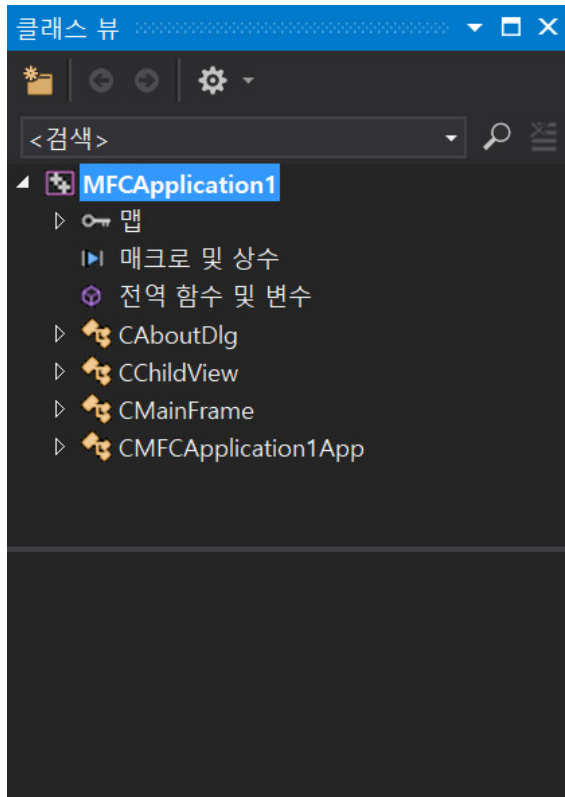
- 솔루션 탐색기 (File View) 화면



- 소스 파일: *.c, *.cpp
- 헤더 파일: *.h
- 자원 파일: *.ico, *.rc

MFC 응용 프로그램 생성

- Class View 화면

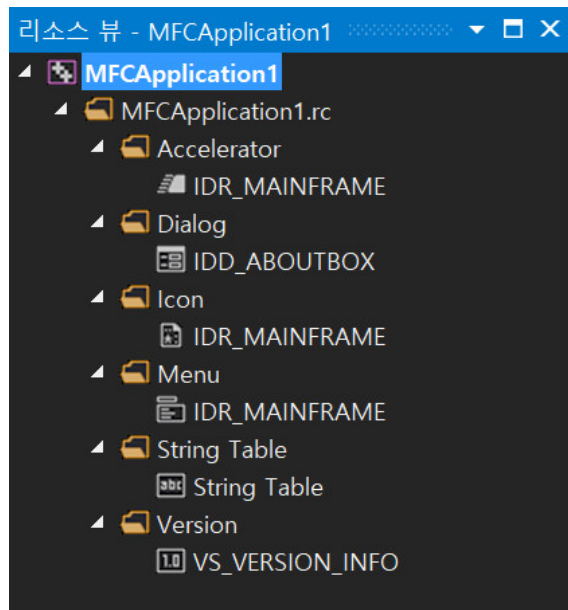


- 해당 class의 member 함수와 변수를 리스트
- 해당 class를 더블 클릭하면 정의부분(class header) 표시
- 해당 class를 선택 후 우측 버튼을 클릭하면 각종 option 설정 menu 표시
- 해당 항목(변수, 함수)를 클릭하면 해당 항목이 설정되어 있는 source code로 이동

MFC 응용 프로그램 생성

- Resource View 화면

- Resource : 윈도우 프로그램을 만들 때 필요한 여러 자원



- Accelerator (hot key 정의) 키 값 정의
- Dialog (대화상자) 자원
- Icon 자원
- Menu 자원
- String(문자열) Table 자원
- Toolbar 자원

MFC 주요 특징

- 빠른 Window 프로그램 작성
 - Library 재사용, AppWizard, ClassWizard, 인쇄 기능 지원, 툴바와 상태바 처리, 데이터베이스(DB) 지원, OLE와 ActiveX 등
 - API를 직접 사용해서 구현할 경우 복잡도가 높은 부분을 MFC를 이용하면 쉽게 구현
- API 기반 program과 대등한 속도를 가짐
 - API 함수를 직접 호출: Ex) ReleaseCapture();
 - inline 함수의 활용: 자주 호출되어야 하는 함수의 경우
- Code 크기 증가를 최소화
 - 동적(Dynamic) library 활용
- C++ 언어를 이용하여 기존의 C 언어에 비해 API를 좀 더 편하게 사용할 수 있음
 - Ex) 오버로딩(Overloading) 및 디폴트 인자



C++?

CLASS?

Q & A